Disease Network Constructor: a Pathway Extraction and Visualization

Mohammad Golam Sohrab[†], **Khoa N. A. Duong**[†], **Goran Topić**[†],
Masami Ikeda[†], **Nozomi Nagano**[†], **Yayoi Natsume-Kitatani**[‡],
Masakata Kuroda[‡], **Mari Nogami Itoh**[‡], **Hiroya Takamura**[†]

[†]Artificial Intelligence Research Center (AIRC),
National Institute of Advanced Industrial Science and Technology (AIST), Japan
[‡]National Institutes of Biomedical Innovation, Health and Nutrition (NIBIOHN), Japan
{sohrab.mohammad, goran.topic, ikeda-masami, takamura.hiroya
n.nagano}@aist.go.jp, {natsume, m-kuroda, mari}@nibiohn.go.jp

Abstract

We present Disease Network Constructor (DNC)[1], a system that extracts and visualizes a disease network, in which nodes are entities such as diseases, proteins, and genes, and edges represent regulation relation. We focused on the disease network derived through regulation events found in scientific articles on idiopathic pulmonary fibrosis (IPF). The front-end web-base user interface of DNC includes two-dimensional (2D) and 3D visualizations of the constructed disease network. The back-end system of DNC includes several natural language processing (NLP) techniques to process biomedical text including BERT-based tokenization on the basis of Bidirectional Encoder Representations from Transformers (BERT), flat and nested named entity recognition (NER), candidate generation and candidate ranking for entity linking (EL) or, relation extraction (RE), and event extraction (EE) tasks. We evaluated the end-to-end EL and end-to-end nested EE systems to determine the DNC's back-end implementation performance. To the best of our knowledge, this is the first attempt that addresses neural NER, EL, RE, and EE tasks in an end-to-end manner that constructs a pathway visualization from events, which we name `Disease Network Constructor`.
The demonstration video can be accessed from https://youtu.be/rFhWwAgcXE8. We release an online system for end users and the source code is available at https://github.com/aistairc/PRISM-APIs/.

1 Introduction

In the human body, various substances (entities) such as proteins and compounds interact and regulate each other, forming huge pathway networks. Such interactions and regulations can be considered as biochemical events. In a disease state, the status of such biochemical events are considered different from those in the healthy state. In order to identify specific substances that can be drug targets in the disease, automatic extraction and visualization of a disease network from scientific articles will be beneficial. The visualization of phenomena and inter-molecular relationships can, for example, make it easier to notice central regulatory molecules, leading to the discovery of drug targets. In this work, we present a system called disease network constructor (DNC) that extracts and visiualizes a disease netowrk. We focus on idiopathic pulmonary fibrosis (IPF), which is a severe chronic fibrosis interstitial lung disease, the causes of which remain unclear (Raghu et al., 2011); thus, a deeper understanding of the disease network is urgently needed. DNC is capable of 3D network drawing, and such 3D visualization can help in understanding diseases such as IPF, where complex factors are entangled and multi-level phenomena are involved.

The task formulation of DNC involves several natural language processinbg (NLP) techniques. DNC is mainly composed of five core models: a Bidirectional Encoder Represenatation from Transformers (BERT)-based **masked language model** (Devlin et al., 2019), **named entity recognition (NER) model** (Sohrab and Miwa, 2018) that enumerates all possible spans as potential entity mentions and classifies them into entity types, **entity linking (EL) model** (Sohrab et al., 2020a) that executes candidate generation and candidate ranking, **relation extraction (RE) model** (Sohrab et al., 2020b), and **event extraction (EE) model** (Trieu et al., 2020). DNC provides a web-based user interface to facilitate the end-to-end process of neural EL and deep EE on the basis of these five models without any training required by end users. The interface visualizes the 2D and 3D networks on the

---

[1]DNC is publicly available at https://biomed-text.airc.aist.go.jp/disease_network/

basis of output of EL to EE.

2 DNC: Back-end System

The BERT-based back-end system of DNC is built upon four layers:

- NER that uses a contextual neural exhaustive approach to extract mentions, entities, and triggers in text.

- EL that normalizes every detected mention by assigning it an ID in the target knowledge base[2].

- RE that extracts all possible role pairs (trigger–trigger and trigger–entity pairs) given detected triggers and entities and assigns a role type to each pair.

- EE that enumerates all legal combinations of role pairs to construct event candidates for each trigger.

We employ the modeling of deep EE (Trieu et al., 2020) on the basis of entity, relation, and event over the IPF dataset (Nagano et al., 2023), which is a manually annotated corpus of IPF-related literature. We further extend the end-to-end deep EE model by leveraging the EL (Sohrab et al., 2020a) model to construct a disease network. Figure 1 shows an overview of DNC workflow.

2.1 BERT Layer

To preprocess a given text, we use BERT's tokenizer to remove special characters and redundant whitespaces, and then split the text into sub-words. A BERT-based pre-trained language model is then used to assign contextual representations to each sub-word.

2.2 Named-entity-recognition Layer

The NER layer assigns entity or trigger types to overlapping text spans by enumerating all possible mention spans on the basis of the same idea as the span-based model (Sohrab and Miwa, 2018; Sohrab et al., 2020b).

2.3 Entity-linking Layer

The EL, or entity normalization, layer, receives the detected mentions $M = \{m_1, m_2, \ldots, m_n\}$ from

the above NER, where m_i denotes the i-th mention and n denotes the total number of extracted mentions. We address the EL in which detected mentions are mapped to the corresponding concept unique identifiers (CUIs) $C = \{c_1, c_2, \ldots, c_n\}$ by leveraging candidate generation and candidate ranking. We use the output of mention extraction as an input to the candidate generation model where we generate a list of k potential CUI candidates for each extracted mention ($k = 50$ in this study). The potential candidates are then fed to the candidate ranking model to select the best candidate for each extracted mention. Our EL layer is based on the EL system of Sohrab et al. (2020a).

2.4 Relation-extraction Layer

The detected mentions and triggers from the NER layer are then fed into the RE layer to assign a role type such as Cause, Cue, Participant, Theme, etc. or no role type to the trigger-argument pairs. The RE layer enumerates all trigger-arguments (trigger-trigger and trigger-entity) to assign a role type.

2.5 Event-extraction Layer

The EE layer receives the detected entities/triggers and the predicted role pairs from the previous layers and enumerates all legal combinations of role pairs to construct event candidates for each trigger. The event candidates include those with and without arguments. Each event candidate is then classified on the basis of whether it is an valid event. Extracting event modifications, such as speculation or negation, is also included in this layer. We describe the event structure to construct event candidates in Section 3.

3 Disease Network Constructor

DNC provides a graph of disease network from the event statistics of IPF. The graph is generated by first applying EL and EE to each input text, then repeatedly collapsing regulation events and their consequents, marking the resultant event with the sign of the regulation event (positive or negative). The resulting graph represents entities as nodes, and regulated events as edges.

We define a "regulation events" as any events with one of the following types: `Positive_regulation`, `Negative_regulation`, or `Regulation`. `Regulation` describes a regulation event for which it is not clear whether its effect is positive
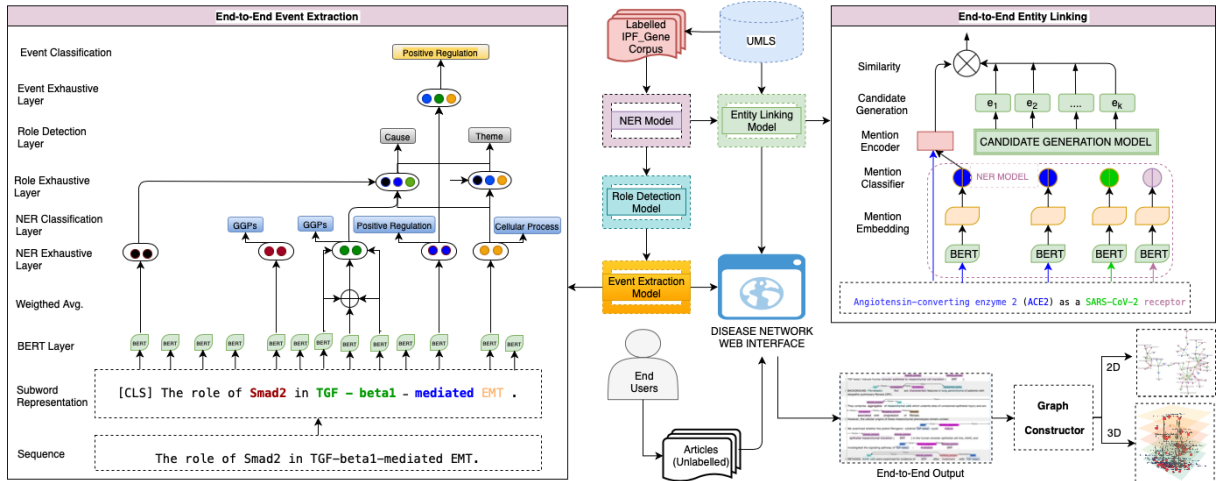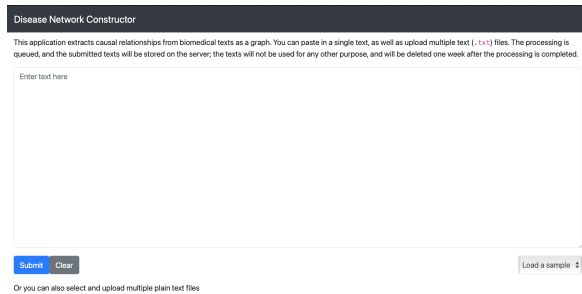
Figure 1: Workflow of DNC.

Figure 2: Web-based user interface of DNC.

or negative, and a regulation sign R is defined as $+1$, -1, and 0, respectively. A regulation event's `Disorder` or `Cause` roles are considered antecedents, and denoted as A; `Theme` roles are consequents, denoted as C. The application collapses regulation events in the following manner.

For each non-negated event E_0 of type T_0 and regulation sign R_0, for each antecedent-consequent pair (A_0, C_0) where A_0 is an entity (or (null), if E_0 has no antecedents): If C_0 is also an entity, we generate a `Direct_regulation`(A_0, C_0, R_0) edge for E_0. If C_0 is a non-regulation event of type T_1, for each of its `Theme` arguments C_1, we generate a $T_1(A_0, C_1, R_0)$ edge for E_0. If C_0 is a regulation event, the data (A_0, E_0, C_0) will be remembered as an uncollapsed regulation link.

After each event is processed as above, we iteratively collapse uncollapsed regulation links until it is no longer possible. We look for an uncollapsed regulation link (A_0, E_0, C_0), such that there exist edges $T_e(A_e, C_e, R_e)$ generated for event C_0. Those edges are deleted, and new edges $T_e(A_0, C_e, R_0 * R_e)$ is generated for E_0. Finally,

any edges with a null source is removed.

Intuitively speaking, each regulation event is "folded into" its consequent as its sign. For example, a "negative regulation" of a "gene expression" becomes a "negatively regulated gene expression" edge, connecting the cause of the "negative regulation" with the theme of the "gene expression". If there is a chain of multiple regulation events, their signs interact: a "negative regulation" of a "negative regulation" of a "gene expression", for example, becomes a "positively regulated gene expression" edge.

This algorithm results in many events not being included on the graph. Since nodes are entities, any regulation event the antecedents of which are not entities are ignored. Similarly, any regulation events that lacks an antecedent or a consequent, as well as any events that do not have a `Theme`, will not be included, as there cannot be an edge without both a source and a target node. Any negated events, and any entities that do not participate in at least one edge are also left out.

3.1 DNC: User Interface

Figure 2 shows a user input interface of DNC. For a given text or single or multiple documents from users or a sample text from the provided list, DNC constructs the graph of regulated events on the basis of EL and EE results and visualizes it in 2D and 3D. The user interface also enables visualization of an already pre-computed disease network by uploading an exported `.json` or `.tgz` file.
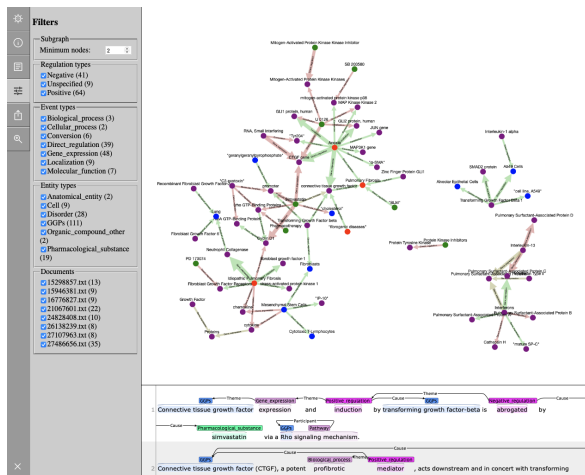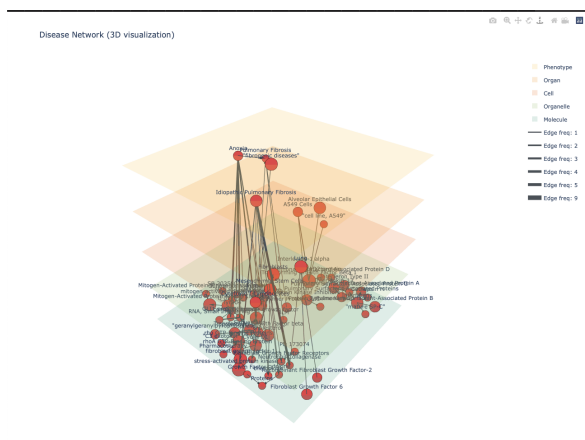
Figure 3: 2D graph produced with DNC.

Figure 4: 3D graph produced with DNC.

Information	IPF Dataset
#Documents	150
#Entity types	15
#Event types	13
#Mentions	12868
#Mentions linked to UMLS CUIs	12148
#Mentions linked to NULL	720
#Entities + Triggers	13049
#Relations	6685
#Events	4899
#Modalities	707

Table 1: Statistics of IPF dataset (Nagano et al., 2023).

Data	P	R	F1 (%)
MD	$86.12_{\pm1.35}$	$87.13_{\pm1.71}$	$86.61_{\pm1.11}$
EL	$72.54_{\pm2.34}$	$59.44_{\pm3.83}$	$65.30_{\pm2.98}$

Table 2: EL performance across 10-fold CV on IPF dataset.

3.2 2D Disease Network

The 2D network based on Cytoscape.js[3] is used to visualize the 2D network graph. Figure 3 shows a 2D graph of a disease network produced with DNC where ten documents or PubMed abstracts on IPF datae are loaded through the user input interface.

3.2.1 2D Graph Features

In the 2D graph, the edges are colored on the basis of whether they represent a positive or negative regulation, or if the regulation type is unknown. Node colors show their types. Node names are normalized where possible; where EL has not succeeded, mentions are grouped into nodes by literal text of the mention, displayed in quotes. In case the mentions represented by an edge or a node do not all have the same type, the color and/or label reflect that of the majority of mentions.

Besides the standard features of zooming and panning, any edge or node can be selected both for better visibility in the graph and displaying more detailed information in a side panel. Nodes can also be selected from an alphabetical list displayed in the info panel when no selection is made. The info panel also enables the selection of edges and nodes in the neighborhood of the selected element and shows the list of all text mentions that the selected element represents. Clicking on a mention displays the EE result in a brat (Stenetorp et al., 2012) visualization panel, enabling easy checking of the context (see the bottom part of Figure 3).

The graph can be filtered by event and entity types, regulation sign, and by the minimum size of connected subgraphs to be displayed. The zoom button enables quick focus on the selected element as well as an overview of the entire graph. The graph can be exported in several formats: tarball (containing the Cytoscape.js JSON representation of the graph as well as input texts and their EE results, which can be later uploaded to the web app to view without having to re-analyze the documents), JSON only (which can also be uploaded to view, though some features will not be available) of both the full graph and its current filtered state, as well as PNG and SVG images.

3.3 3D Disease Network

The web-base user interface also has a 3D graph function, the main advantage of which is that the nodes are split into layers, representing phenotypes,

---

[3]https://js.cytoscape.org

Task	(%)	10-fold Cross-Validation of End-to-End Entity Linking									
		$Fold_1$	$Fold_2$	$Fold_3$	$Fold_4$	$Fold_5$	$Fold_6$	$Fold_7$	$Fold_8$	$Fold_9$	$Fold_{10}$
MD	P	87.84	87.29	84.19	85.47	87.64	85.54	84.87	85.08	85.57	87.67
	R	88.56	89.17	85.62	86.02	86.57	87.11	87.03	87.94	89.38	83.87
	F	88.21	88.22	84.90	85.75	87.10	86.31	85.93	86.49	87.44	85.73
EL	P	71.96	71.47	75.29	74.31	70.40	68.59	71.52	75.78	74.57	71.48
	R	63.86	61.68	62.84	60.17	53.11	56.66	57.12	59.55	64.29	55.14
	F	67.67	66.22	68.51	66.50	60.54	62.06	63.51	66.69	69.05	62.26

Table 3: EL performances across 10-fold CV. MD indicates mention detection.

Task	P	R	F (%)
NER	$84.77_{\pm1.56}$	$82.05_{\pm2.65}$	$83.37_{\pm1.73}$
RE	$58.62_{\pm4.20}$	$59.49_{\pm3.97}$	$58.95_{\pm3.09}$
EE	$51.55_{\pm4.17}$	$40.10_{\pm4.17}$	$45.08_{\pm4.12}$
ME	$51.59_{\pm14.97}$	$26.09_{\pm10.72}$	$34.24_{\pm12.06}$

Table 4: EE performances across 10-fold of CV on IPF dataset. ME indicates modality extraction.

organs, cells, organelles, and molecules. Figure 4 shows a 3D graph of a disease network produced with DNC.

4 Experimental Settings

In this section, we evaluate the DNC system based on the IPF dataset (Nagano et al., 2023).

4.1 Datasets

We conduct experiments on the IPF dataset which includes 150 abstracts of IPF-related scientific literature where entity, relation, and event information are manually annotated (Nagano et al., 2023). Table 1 shows the statistics of the IPF dataset, which is split into training set and test set for 10-fold cross-validation (CV) in this work. The IPF dataset is randomly divided into 10 folds, each turn, one data fold is used for testing and the remaining folds are used for training.

Moreover, to address IPF-related networks, this dataset includes entity normalization with concept unique identifiers (CUIs) assigned to entities. The UMLS version 2017AA [4] is used to assign the CUIs of entities. It contains 2.1M unique CUIs which covers 100% of entities in the IPF dataset. As shown in Table 1, the IPF dataset includes 12,319 mentions among which 12,148 and 720

mentions are respectively present and absent in the UMLS. Therefore, the entity coverage ratio of the IPF dataset over the UMLS is around 94.3%.

4.2 Implementation Details

We train the EL and EE models on the pre-trained BERT model and use the pre-trained PubMed-BERT(Gu et al., 2020) for end-to-end EL task. We employ the pre-trained SciBERT(Beltagy et al., 2019) model to address the end-to-end event extraction task. We optimize the end-to-end EL and end-to-end EE models using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 3e-5. We train our EL and EE models with 100 epochs and a mini-batch size of 16 on a single graphics processing unit (GPU) with half precision enabled.

5 Results

Table 2 shows the end-to-end EL performances based on the IPF dataset, with the mean scores of precision (P), recall (R), and F-score (F) over the 10-fold CV. The $\pm(.)$ subscript indicates the standard deviation of variation of a set of 10-fold CV scores. Table 3 shows the 10-fold CV end-to-end EL performances over the IPF dataset. The overall performances in Table 2 and the consistent performances over each fold in Table 3, suggest that MD and EL perform well on the IPF dataset.

Table 4 shows the end-to-end EE performances based on the IPF dataset where the end-to-end deep event extraction model is simultaneously trained for entity/trigger, role, and event detection. Since the EE model follows the end-to-end manner, therefore it is noticeable that the model performances are decreased from the NER layer to the modality extraction (ME) layer where each layer error is propagated to the next layer, making the task challenging for the following layers. Table 5 shows the results of the 10-fold CV of the end-to-end deep EE. ME does not perform well compared with the

---

[4]https://www.nlm.nih.gov/pubs/techbull/mj17/mj17_umls_2017aa_release.html

	(%)	Fold$_1$	Fold$_2$	Fold$_3$	Fold$_4$	Fold$_5$	Fold$_6$	Fold$_7$	Fold$_8$	Fold$_9$	Fold$_{10}$
	10-fold Cross-Validation of End-to-End Deep Event Extraction										
NER	**P**	85.46	84.97	82.74	84.94	88.28	85.82	84.05	84.47	83.49	83.48
	R	84.01	83.04	81.29	80.71	82.20	82.73	82.55	84.19	84.42	75.35
	F	84.72	83.99	82.01	82.77	85.13	84.25	83.29	84.33	83.95	79.21
RE	**P**	61.13	56.76	60.64	61.10	55.77	62.54	64.53	58.75	54.18	50.83
	R	62.05	59.95	57.26	57.23	59.61	60.55	60.06	58.83	67.54	51.86
	F	61.59	58.31	58.91	59.10	57.63	61.53	62.21	58.79	60.12	51.34
EE	**P**	52.99	52.19	54.80	50.11	50.93	41.48	52.55	51.42	57.74	51.25
	R	41.03	39.49	43.51	35.82	39.25	32.56	41.26	43.56	47.01	37.47
	F	46.24	44.96	48.51	41.78	44.34	36.48	46.22	47.16	51.82	43.29
ME	**P**	48.65	64.86	73.08	43.75	58.33	43.75	50.01	19.35	64.10	50.01
	R	32.14	33.80	32.26	17.28	25.61	18.75	27.85	10.01	46.55	16.67
	F	38.71	44.44	44.76	24.78	35.59	26.25	35.77	13.19	53.94	25.01

Table 5: 10-fold cross validation (CV) of end-to-end deep event extraction. ME indicates modality extraction.

other extractions due to insufficient gold data as in Table 1.

6 Related Work

Recent successes in neural networks have shown impressive performance gains on many NLP applications including NER (Lu and Roth, 2015; Ma and Hovy, 2016; Muis and Lu, 2017; Katiyar and Cardie, 2018; Sohrab et al., 2019b; Sohrab and Bhuiyan, 2021), EL (Gupta et al., 2017; Sohrab et al., 2019a), RE (Christopoulou et al., 2019; Jia et al., 2019), and EE (Feng et al., 2016). In contrast, other approaches have emphasized end-to-end EL (Kolitsas et al., 2018), end-to-end RE (Miwa and Bansal, 2016) or even end-to-end EE (Trieu et al., 2020) to facilitate biomedical information extraction tasks. There have been no studies on an all-in-one neural end-to-end approach to facilitate biomedical research, especially to construct disease network pathways that visualize the events along with entity normalization. We addressed this gap by introducing two end-to-end approaches: EL and deep EE to construct a disease network based on IPF, hoping that the DNC can bring insights in making scientific discovery.

Current NLP techniques often use an event representation data format called the "standoff format" to represent their results. Spranger et al. (2015) proposed and discussed a software scheme to convert NLP event representations to standard biomedical pathway data formats (SBML and BioPAX). Apart from neural end-to-end modeling, we integrated brat visualization panels for event representation of the context.

There are several web-based tools exist that support the retrieval of biomedical information using text mining. Sohrab et al. (2020a) introduced BEN-NERD a web-based workflow of NER and EL for NLP research that addresses COVID-19 research. Huang et al. (2021) addressed document-level EE, for extracting entity-centric information such as entity types and entity relations, which is a key to automatic knowledge acquisition from text corpora for various domains. Sohrab et al. (2022) presented an effective web application by addressing entity detection, EL without context using knowledge base application programming interfaces (API), generative RE, and text classification approaches in a pipeline manner for automatic data curation in the biomedical domain. The advantage of this approach is that it can output important fields in a data format that is needed by intended users.

Li et al. (2022) proposed pubmedKB, a web server designed to extract and visualize semantic relationships between four biomedical entity types: variants, genes, diseases, and chemicals. pubmedKB uses state-of-the-art NLP techniques to extract semantic relations from a large number of PubMed abstracts. Deng et al. (2021) addressed an extraction of gene-disease association using a BERT-based language model. Xing et al. (2018) proposed a pipeline based approach to extract the relation between gene-phenotype from biomedical literature.

Many works have shown considerable attention to boost the EE performances. Previous neural models on flat or non-nested EE have been mainly focused on event trigger and argument de-

tection (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Sha et al., 2018). Besides, deep neural networks including recurrent and convolutional neural networks (CNNs) have boosted EE performance (Björne and Salakoski, 2018; Nguyen and Nguyen, 2019). These models show better performance than traditional hand-crafted feature-based approaches (Björne and Salakoski, 2013; Miwa and Ananiadou, 2013; Yang and Mitchell, 2016). In addition, there are a few end-to-end models (Yang and Mitchell, 2016; Nguyen and Nguyen, 2019) to extract flat events on flat entities; none of these models can treat nested events on nested entities that may further overlap with event triggers. In contrast, Trieu et al. (2020) introduced an end-to-end neural nested EE model which detects nested entities and triggers, roles, nested events; and achieved the new state-of-the-art performance on seven biomedical nested event extraction tasks. In our DNC, we employ the modeling of Deep EE (Trieu et al., 2020) to detect the flat and nested events over the IPF dataset.

7 Conclusion

We present Disease Network Constructor (DNC) to address end-to-end EL and end-to-end deep EE in order to identify and visualize the specific substances (such as proteins) that work differently from those in the healthy state of human bodies. DNC provides an interactive web-based user interface `https://biomed-text.airc.aist.go.jp/disease_network/` for enabling real-time visualization and extracting graph information in different formats for end users. We will continue to improve DNC as well as implement new 2D and 3D graph functions to facilitate biomedical research. Moreover, the applicability of this system can be extended to lung diseases such as COVID-19 because some entities and events of the IPF dataset are also related to such diseases.

Acknowledgements

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciB-ERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Jari Björne and Tapio Salakoski. 2013. TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 16–25, Sofia, Bulgaria. Association for Computational Linguistics.

Jari Björne and Tapio Salakoski. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, pages 98–108, Melbourne, Australia. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Chuan Deng, Jiahui Zou, Jingwen Deng, and Mingze Bai. 2021. Extraction of gene-disease association from literature using biobert. *The 2nd International Conference on Computing and Data Science*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark. Association for Computational Linguistics.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multi-scale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Peng-Hsuan Li, Ting-Fu Chen, Jheng-Ying Yu, Shang-Hung Shih, Chan-Hung Su, Yin-Hung Lin, Huai-Kuang Tsai, Hsueh-Fen Juan, Chien-Yu Chen, and Jia-Hsin Huang. 2022. pubmedKB: an interactive web server for exploring biomedical entity relations in the biomedical literature. *Nucleic Acids Research*, 50(W1):W616–W622.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations, ICLR 2019, New Orleans, USA*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Conference on Empirical Methods in Natural Language Processing*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Sophia Ananiadou. 2013. NaCTeM EventMine for BioNLP 2013 CG and PC tasks. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 94–98, Sofia, Bulgaria. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark. Association for Computational Linguistics.

Nozomi Nagano, Narumi Tokunaga, Masami Ikeda, Inoura Hiroko, Duong A. Khoa, Makoto Miwa, Mohammad Golam Sohrab, Goran Topić, Mari Nogami-Itoh, and Hiroya Takamura. 2023. A novel corpus of molecular to higher-order events that facilitates the understanding of the pathogenic mechanisms of idiopathic pulmonary fibrosis. *Scientific Reports*, 13(1, 5986).

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6851–6858.

Ganesh Raghu, Harold R Collard, Jim J Egan, Fernando J Martinez, Juergen Behr, Kevin K Brown, Thomas V Colby, Jean-François Cordier, Kevin R Flaherty, Joseph A Lasky, David A Lynch, Jay H Ryu, Jeffrey J Swigris, Athol U Wells, Julio Ancochea, Demosthenes Bouros, Carlos Carvalho, Ulrich Costabel, Masahito Ebina, David M Hansell, Takeshi Johkoh, Dong Soon Kim, Talmadge E King Jr, Yasuhiro Kondoh, Jeffrey Myers, Nestor L Müller, Andrew G Nicholson, Luca Richeldi, Moisés Selman, Shandra

L Protzko Holger J Schünemann Rosalind F Dudden, Barbara S Griss, and ATS/ERS/JRS/ALAT Committee on Idiopathic Pulmonary Fibrosis. 2011. An official ats/ers/jrs/alat statement: idiopathic pulmonary fibrosis: evidence-based guidelines for diagnosis and management. *American journal of respiratory and critical care medicine*, 183(6):788–824.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Mohammad Golam Sohrab and Md. Shoaib Bhuiyan. 2021. Span-based neural model for multilingual flat and nested named entity recognition. In *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*, pages 80–84.

Mohammad Golam Sohrab, Khoa Duong, Makoto Miwa, Goran Topić, Ikeda Masami, and Hiroya Takamura. 2020a. Bennerd: A neural named entity linking system for covid-19. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 182–188, Online. Association for Computational Linguistics.

Mohammad Golam Sohrab, Khoa N.A. Duong, Ikeda Masami, Goran Topić, Yayoi Natsume-Kitatani, Masakata Kuroda, Mari Nogami Itoh, and Hiroya Takamura. 2022. BiomedCurator: Data curation for biomedical literature. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 63–71, Taipei, Taiwan. Association for Computational Linguistics.

Mohammad Golam Sohrab, Anh-Khoa Duong Nguyen, Makoto Miwa, and Hiroya Takamura. 2020b. mg-sohrab at WNUT 2020 shared task-1: Neural exhaustive approach for entity and relation recognition over wet lab protocols. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 290–298, Online. Association for Computational Linguistics.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Golam Sohrab, Minh Thang Pham, Makoto Miwa, and Hiroya Takamura. 2019a. A neural pipeline approach for the PharmaCoNER shared task using contextual exhaustive models. In *Proceedings of the 5th Workshop on BioNLP Open Shared Tasks*, pages 47–55, Hong Kong, China. Association for Computational Linguistics.

Mohammad Golam Sohrab, Pham Minh Thang, and Makoto Miwa. 2019b. A generic neural exhaustive approach for entity recognition and sensitive span detection. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019, Bilbao, Spain, September 24th, 2019*, volume 2421 of *CEUR Workshop Proceedings*, pages 735–743. CEUR-WS.org.

Michael Spranger, Sucheendra Palaniappan, and Samik Ghosh. 2015. Extracting biological pathway models from NLP event representations. In *Proceedings of BioNLP 15*, pages 42–51, Beijing, China. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Hai-Long Trieu, Thy Thy Tran, Khoa N A Duong, Anh Nguyen, Makoto Miwa, and Sophia Ananiadou. 2020. DeepEventMine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917.

Wenhui Xing, Junsheng Qi, Xiaohui Yuan, Lin Li, Xiaoyu Zhang, Yuhua Fu, Shengwu Xiong, Lun Hu, and Jing Peng. 2018. A gene–phenotype relationship extraction pipeline from the biomedical literature using a representation learning approach. *Bioinformatics*, 34(13):i386–i394.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.